WEEKLY REPORT – Sindhuja Chaduvula

Lanelet 2 Overview

Introduction

A high definition map format for the future of automated driving refers to a modern mapping framework specifically designed for the challenges and requirements of Autonomous driving. Lanelet 2 is the name of the specific mapping framework being referenced to.
High-definition map framework – Traditional maps, like the one you might use for everyday navigation, are not detailed enough for autonomous vehicles. Automated or self driving cars require very high-definition maps that provide granular details, down to the specific lane markings, traffic light positions, curb height, and more. When we say high-definition in this context, we are talking about the level of detail and precision, not the visual resolution. The goal of Lanelet2 is not only to be usuable for typical, isolated applications such as localization or motion planning, but for various potential applications of maps for highly automated driving. The reason for lanelet2 is not only that it compensates for the inadequacies of the sensors, but also that maps are important for providing information about regions that cannot be observed by sensors, whether due to occlusion or insufficient sensor range. In addition a correct interpretation of the observed data is requied, which copes with the uncertainities of the sensors.

- Maps provide the reliable information needed to understand a scene correctly.
- Maps allow to transfer knowledge from previous journeys and thus represent an additional level of redundancy.
- Information about the complex environment of a vehicle such as bicycle lane and sidewalks must also be made available and should therefore be provided by a map.

Drawbacks in Existing Map Formats

Open Street Map(osm) – Roads are represented by an imaginary center line. This framework is developed for uniform description of the road and its surroundings in driving simulators, but is now also used for HAD. This map format with lane-level accurate information have also been successfully used for driving on motorways.

Liblanet – This format made a simple, explicit representation of the lanes possible. Traffic rules are represented by so called Regulatory elements. It dealt with topic of localization, Map Generation, System Architecture, Motion Planning Prediction. It focuses more on the road network. It needs detailed information about lanes and their environment. Also needs direct access to individual physical elements of a map.

OSM provides a highly versatile and globally-available mapping solution, Lanelet augments this with the specific, detailed data necessary for autonomous vehicles to operate safely and effectively. Libraries built for working with Lanelet data, such as liblanelet, are thus specialized tools that address the shortcomings of OSM in the context of autonomous driving but may not be as broadly applicable for other mapping needs.

The article highlights the limitations of existing formats like OpenDRIVE, emphasizing their absence of open-access libraries and challenges in deciphering the data. Additionally, it explores map models designed for particular uses, such as the Unified Map tailored for trajectory planning, underscoring that these models are confined to specific use cases.

The article reviews 32 studies related to highly automated driving from the past five years that specifically employ high-resolution maps. Most of these studies focus on localization, and they either craft their map formats or modify existing ones, pointing to a deficit in standardized formats. A few adopted OSM, Liblanelet, OpenDRIVE, or HERE maps, showcasing a variety of preferences and the lack of a universally adopted standard.

Essential Criteria for High Level Automated Driving Maps

Road Network - Requires greater accuracy, minimize lane changes, when lane changes are happening find alternative lanes, Behaviour Generation, Right of way rules that is a choice is made between different maneuvers such as overtaking, merging or braking maneuver, prediction of other road users. Types of road users(bus, pedestrian, cyclists).

Lane and Environment – Path planning stresses the importance of lane geometry. Obstacles such as parking vehicles protruding into the lane often have to be avoided without endangering oncoming traffic. Appropriate parking facilities must first be known for reaching a parking space.

Physical Elements – Map should therefore not only contain the derived information, but also allow conclusions to be drawn as to where this information come from. Physical elements need to be stored and referenced in a way that is robust against environmental changes.

Architecture of lanelet2

- Map is divided into a physical layer(contains the usually real observable elements)
- Relational layer(Elements of the physical layer are connected to lanes, areas and traffic rules)
- Topological layer(Elements of the relational layer are combined to a network of potentially possible regions depending on the road user and the situation)

All elements in the map can be described predominantly by a projection onto a flat ground plane. A lanelet map consists of five elements.

- Points and line strings belonging to the physical layer.
- Lanelets, areas and regulatory elements belonging to the relational layer.

Points – Points are the basic element of the map, a single point represent vertical structure such as poles. Points are the only primitives that actually have position information. All other primitives are directly or indirectly composed of points.

Linestrings – Linestrings are an ordered array of two or more points between which linear interpolation takes place. It is used to represent shape of elements in the map. Road markings, curbs facades etc. Can be virtual if they form an implicit border of a lane.

Lanelets - Lanelets define an atomic section of the map in which directed motion takes place. Normal lanes but also pedestrian crossings and rails. Atomic(Currently valid traffic rules do not change within a lanelet, but also that the topological relationships with other lanelets does not change). A lanelet is defined by exactly on line string as the left and exactly one as the right border. Lanelets can also overlap or intersect. The type of the left or right border expresses whether for example lane changes to an adjacent lanelet are possible.

Areas – Sections of the map in which undirected or no movement is possible. Parking areas, squares, green spaces or buildings.

Regulatory elements – Define traffic rules, such as speed limits, priority rules or traffic lights.

Modules
It is important to separate the representation of the elements of a map from their interpretation. Routing, matching and access to the individual physical elements.

Traffic rules – This serves to interpret the rules contained in the map depending on the type of road user and country. It also determines whether lanechanges are possible or whether a certain road user is permitted to enter a lanelet.

Physical – This module allows direct, comfortable access to the elements of the physical layer.

Routing – With the help of traffic rules, routing graphs can be set up to determine the exact route to be driven inclusing possible lane changes, predicting routes and potential conflict points for other road users.

Matching – This module is used to assign lanelets to road user or determine potential positions on the map based on specific observations collected from the sensors.

Projection – Contains functionality to convert global latitude/longitude coordinates to local, metric coordinates. This conversion is important for the seamless integration of external geographic data into the lanelet2 framework.

Input/Output(IO) – The input/output module contains functions for reading and writing map formats in particular OSM format. This module enables Lanelet2 to interface with different file structures, facilitating data import and export.

Together, these modules empower Lanelet2 to handle and comprehend map data efficiently, offering a range of features that cater to different facets of map use in high-level automated driving contexts.

Explore potential prediction or classification models to determine at which timestamps of total simulation time and lanes/positions will collision happen under specific configurations.

When it comes to predicting collisions in a simulated environment like SUMO, a variety of prediction and classification models can be applied.
Linear Regression - These are the simplest form of predictive models which can be applied when the relationship between features (like speed, lane, position) and the target (collision

time) is linear. Performance in SUMO might not be very accurate due to the complex nature of traffic interactions. Generally used as a baseline model.

Decision Trees and Random Forests:
Decision Trees: They split the dataset into subsets based on the feature values. This continues recursively, forming a tree of decisions.
Random Forests: Ensembles of decision trees that aim to improve the accuracy and robustness over a single decision tree. SUMO performance could be effective in capturing complex interactions in traffic and understanding feature importance, e.g., which lanes or positions are more prone to collisions.

Gradient Boosted Machines (GBM): Boosting algorithms that build an additive model in a forward stage-wise manner. XGBoost and LightGBM are popular implementations.
In SUMO, GBMs, especially XGBoost, can handle large datasets and complex interactions, making them a strong candidate for predicting collisions in SUMO simulations.

Neural Networks: Multi-layer architectures that can model complex non-linear relationships. In SUMO, Neural networks, especially recurrent architectures like LSTM or GRU, can be used for time series prediction. They might be particularly useful when considering the sequence of vehicle movements leading up to a collision.

Support Vector Machines (SVM): Used for both classification and regression. SVM tries to find a hyperplane in a multi-dimensional space to separate different classes. In SUMO, SVMs might work for simpler scenarios, but can become computationally expensive with large datasets.

K-Nearest Neighbors (KNN): A non-parametric method that can be used for both classification and regression. It predicts based on the 'k' training samples closest in distance. In SUMO it might be computationally intensive with large datasets but can give insights into similar traffic scenarios leading to collisions.

The attributes associated with vehicle dynamics and the environment within a roundabout simulation

id: A unique identifier for a vehicle or entity within the simulation or dataset. Helps in tracking the specific movement and characteristics of individual vehicles.
x, y: The x and y coordinates of the vehicle. They represent the spatial position of the vehicle in a 2D plane, typically a map or road network.
angle: The orientation of the vehicle with respect to a reference axis. It can indicate the direction in which the vehicle is moving or facing.
type: Specifies the category or classification of the vehicle. For instance, it could be a car, truck, bus, etc.
speed: The current speed of the vehicle, often measured in units like meters per second (m/s) or kilometers per hour (km/h).
pos: Position of the vehicle along its current lane or path. This could be a distance from a reference point like the start of the lane or a junction.

lane: The specific lane in which the vehicle is currently traveling. Roads can have multiple lanes, and this attribute identifies which one the vehicle is in.
time: The timestamp at which the data was recorded. Useful for time-series analysis and understanding vehicle dynamics over a period.
lane_id: A unique identifier for the lane. It helps in mapping which lane (of potentially many) the vehicle is on.
lane_length: The total length of the current lane. Useful for calculations related to how much of the lane a vehicle has covered.
lane_change: Indicates whether the vehicle has changed lanes. It could be a binary flag or could specify the direction of the change (e.g., left, right).
vehicle_density: Represents the number of vehicles within a specific area or length of the road. High density could imply congestion or heavy traffic.
avg_speed_nearby_vehicles: The average speed of vehicles in the vicinity of the current vehicle. This can give insights into traffic flow and potential congestion.
collision: A binary flag or indicator (e.g., 0 or 1) specifying whether a collision has occurred. '1' might indicate a collision, while '0' indicates no collision.

The output performance of a RandomForest classifier on a classification task. With an accuracy of approximately 99.91%, it correctly predicted around 99.91% of the test set labels. The classification report suggests it excels at predicting the majority class (0) but has challenges with the rarer class (1). Specifically, while its precision for class 1 is perfect, its recall is only 0.14, indicating it's missing many true positive instances. This is further reflected in the F1-score of 0.25 for class 1. The data appears imbalanced with 19159 instances of class 0 and only 21 of class 1. In such imbalanced datasets, high accuracy can be misleading. Given the model's challenges with the minority class, considering techniques like oversampling, undersampling, or alternate evaluation metrics could be beneficial for a more balanced performance.